FIG. 1

FIG. 2

Decoding/
Decompression
Module
212

219

Decoder
Module
216

Primary
Channel
112

Assist
Channel
118

Image Extraction
Module
210

Bitmap
Primary
Channel
214

Hardcopy Assist Channel
204

206

208

102

104

114

106

124

126

FIG. 3

Receive A Primary Channel Of Character Data — 302

Identify The Alphabet Of Alphanumeric Symbols That May Appear In A Primary Channel Of A Set Of Electronic Documents — 306

Partition The Identified Alphabet of Alphanumeric Symbols Into A Plurality Of Groups That Minimize Entropy And Maximize Benefit — 308

Assign An Identifier To Each Of The Plurality Of Groups — 310

Develop An Assist Channel By Associating Each Symbol Of The Primary Channel With An Identifier Corresponding To The Group To Which Each Symbol Is Assigned — 314

Compress The Assist Channel — 316

Encode The Assist Channel — 318

Concatenate The Primary Channel With The Encoded Assist Channel — 320

Render The Concatenated Primary Channel And Assist Channel On A Hardcopy Document — 322

Defining

Applying

304

312

DOES TSO"B304Y550

306

→ Define A Graph With A Node For Each Character In The Identified Alphabet
of Symbols And Arcs That Interconnect Each Node — 410

→ Assign Values To The Nodes And Arcs As Follows:

(a) Assign Each Node With The Probability That A Character Will Occur In The Document; and
(b) Assign Each Arc With The Likelihood A Character Will Be Confused With Another Character — 412

→ Initialize The Current Partition To Be The Partition Consisting Of The Partition With
One Group Containing All The Nodes Of The Graph — 414

→ Identify All Possible Changes In The Current Partition Of The Graph By:
1. Forming A New Group Of Any Node From Any Existing Group; and
2. Moving A Node From One Group To Another. — 416

→ For Each Of The Candidates Identified By The Possible Changes Identified At Step 416 Compute:
A ΔBenefit And A ΔEntropy — 418

→ Select The Current Partition To Be That Candidate That
Maximizes The Ratio Of The Change In Benefit To The Change In Entropy
(i.e., Measure Of Reduction In OCR Errors/
Measure Of Size Of Compressed Assist Channel) — 420

→ Reached Desired Benefit Or Entropy? — 422

No → (loops back)

Yes → 310

308

FIG. 4

FIG. 5

Entropy

610

608

606

602

612

616

614

618

604

600

-Benefit

FIG. 6

700

708 a:
(00)
702 714
Athsaide
wgpkmy

710 b:
704
(01)
o n f
716

712 c:
706
(10)
c l u r b v q x -
718

FIG. 7

005450.894ч7560

An assist channel is a compact machine-readable

800

802 804
806 808

FIG. 8

abaaaaaaacaabbacaaaacbaaacaaaacaaabaccaaacaaa

900

902 904
906 908

FIG. 9

1010

FIG. 10

COOL T.O. " Pentagasso

Acquire Binary Image — 1102

Extract Assist Channel From The Binary Image — 1104

Decode The Assist Channel — 1106

Decompress The Assist Channel — 1108

Acquire Next Line Of The Primary Channel And A Corresponding Next Line Of Assist Channel — 1110

Determine A Graph Of The Primary Channel To Define A First Dimension Of A Product Graph — 1112

Determine A Graph Of The Assist Channel To Define A Second Dimension Of The Product Graph — 1114

Determine The Shortest Path Of The Product Graph To Produce An Observed Primary Channel — 1116

Have All Lines Been Processed? — 1118

No

Yes — 1120

Done

FIG. 11

005 TE0 " B9c4745E0

```
begin
  │
  ▼
┌─────────────────────────────────────────┐
│ At Every Pixel Position In The Line Of   │  1202
│ Text Define A Node Of A Graph            │
└─────────────────────────────────────────┘
  │
  ▼
┌─────────────────────────────────────────┐
│ At Each Node Defined At Step 1202,       │  1203
│ Enumerate All Possible Arcs Corresponding│
│ To Templates                             │
└─────────────────────────────────────────┘
  │
  ▼
┌─────────────────────────────────────────┐
│ Compute A Score For Each Arc By Matching │  1204
│ The Character Defined In The Template    │
│ Library With The Character In The Image  │
└─────────────────────────────────────────┘
```

1200 — Define Graph

```
┌─────────────────────────────────────────┐
│ Define A Record With A Cumulative Score  │  1206
│ And A Backpointer For Each Node          │
└─────────────────────────────────────────┘
  │
  ▼
┌─────────────────────────────────────────┐
│ Initialize The Record At Each Node       │  1208
└─────────────────────────────────────────┘
  │
  ▼
┌─────────────────────────────────────────┐
│ Perform A First ("Forward") Pass To      │  1210
│ Compute The Shortest Path For Each Node  │
└─────────────────────────────────────────┘
  │
  ▼
┌─────────────────────────────────────────┐
│ Perform A Second ("Backward") Pass To    │  1212
│ Identify The Shortest Path Of The Graph  │
└─────────────────────────────────────────┘
  │
  ▼
 end
```

1201 — Compute Shortest Path

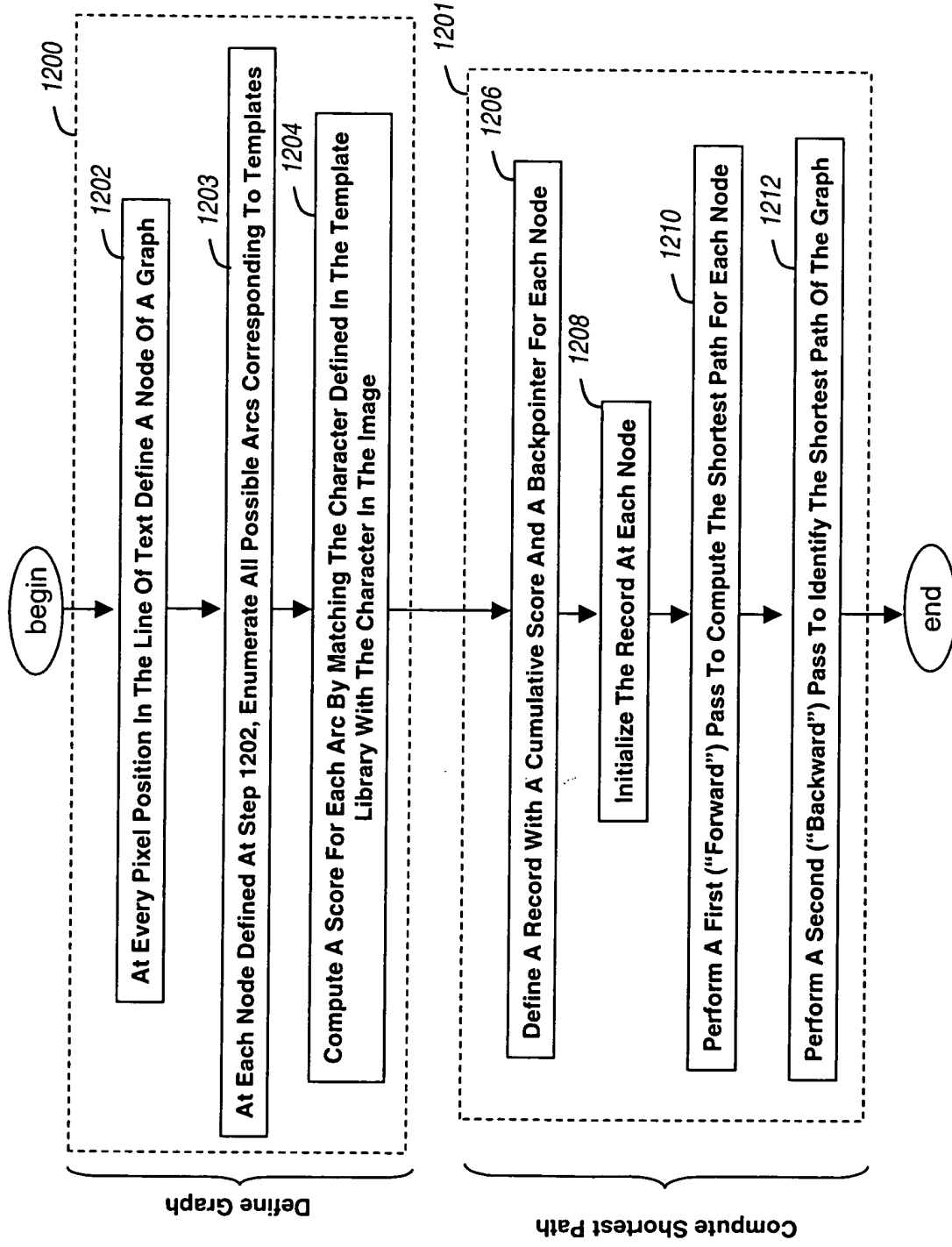FIG. 12
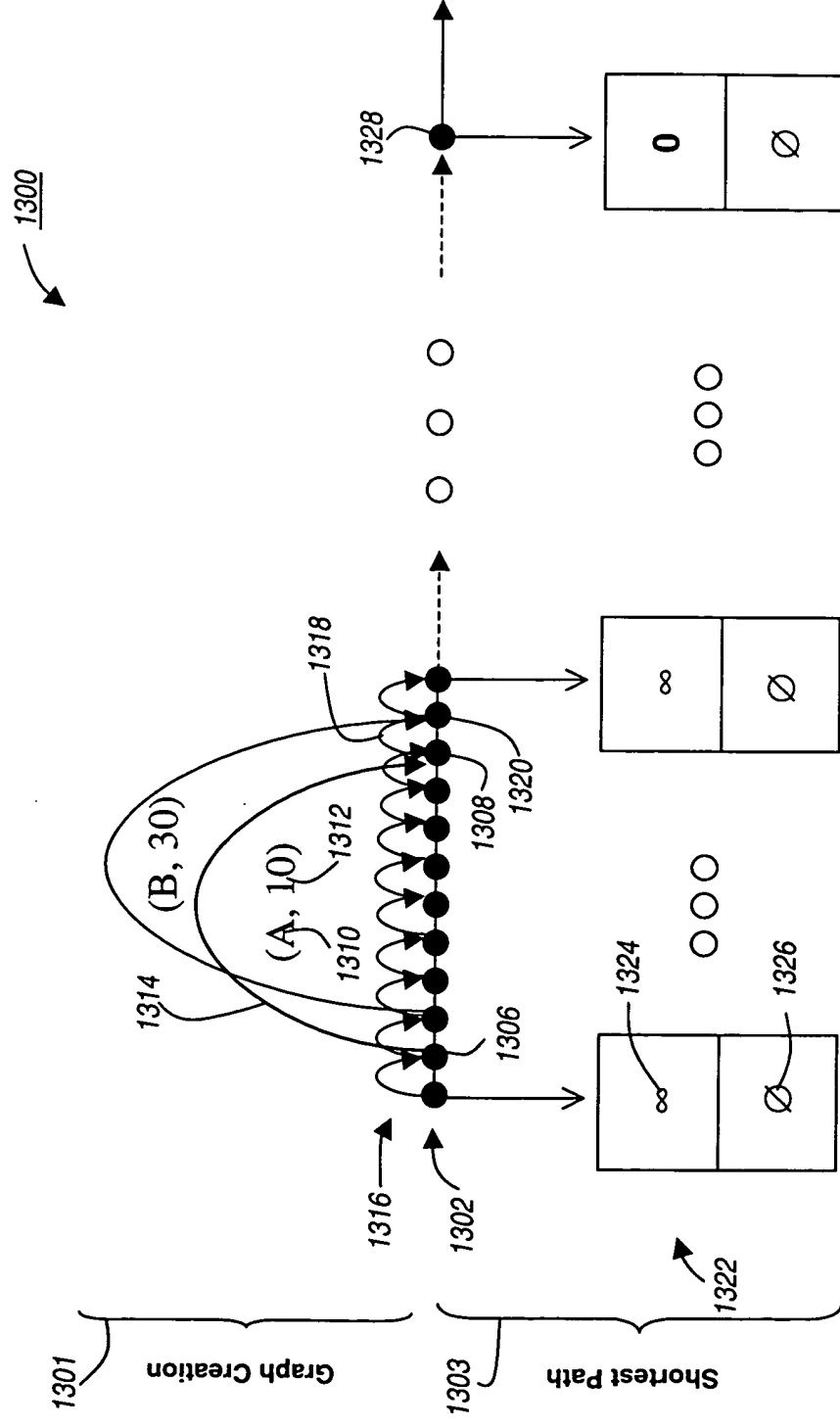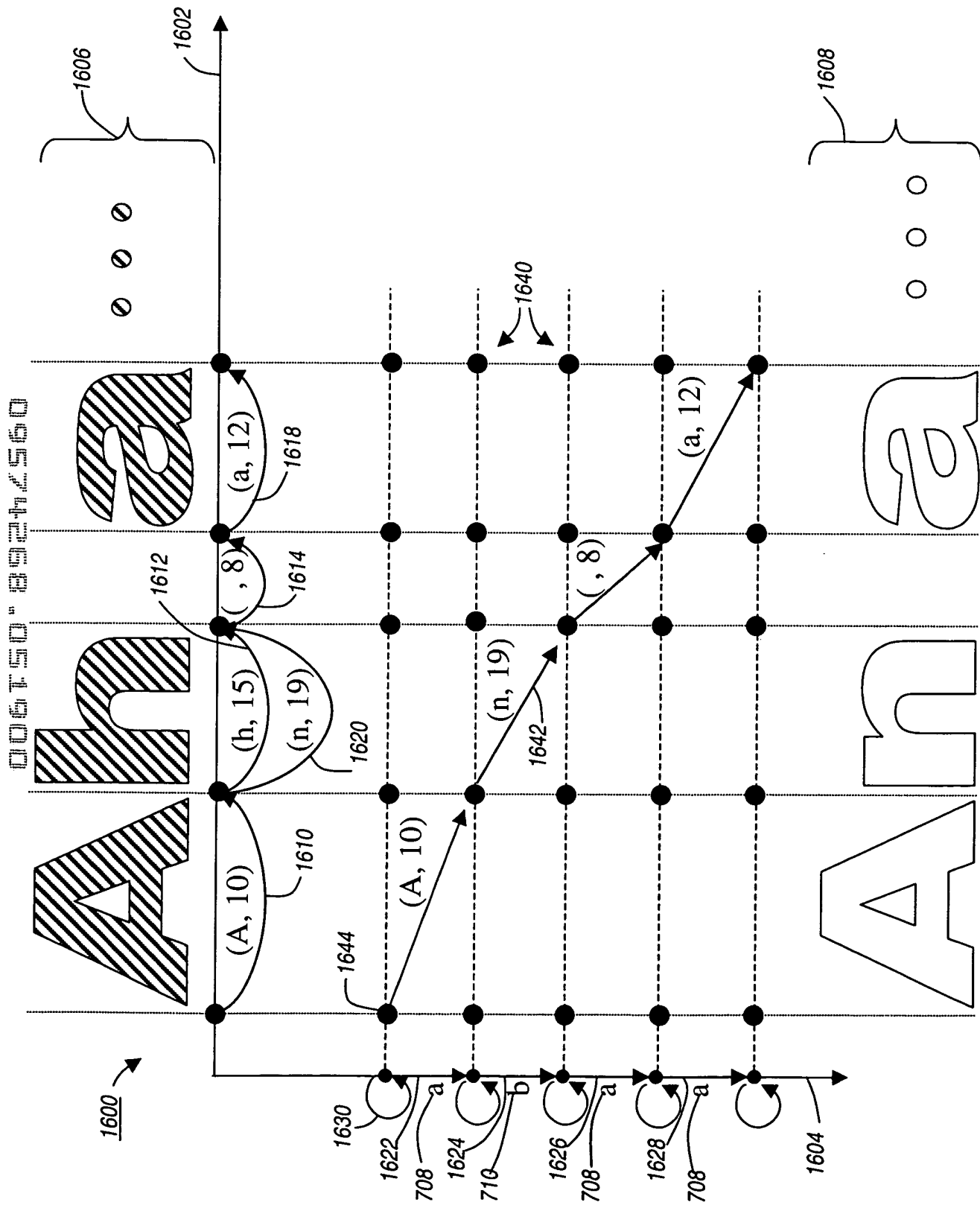
FIG. 13

FIG. 14

FIG. 15

FIG. 16

Receive A Primary Channel Of Symbol Data — 1702

Specify A Column Group Size — 1704

Specify A Number Of Guard Digits Per Column Group — 1706

Determine The Number Of Lines Of Symbol Data In The Primary Channel — 1708

Determine The Lengths Of Each Line Of Symbol Data — 1710

Divide The Primary Channel Of Symbol Data Into Vertical Blocks According To The Specified Column Group Size — 1712

For Each Vertical Block Compute The Specified Number Of Guard Digits — 1714

Assemble And Compress In The Assist Channel: (a) Column Group Size; (b) Number Of Guard Digits Per Column Group; (c) Number Of Lines Of Symbol Data; (d) Vector Of Line Lengths; (e) Guard Digits For Each Column Group — 1716

Encode Assist Channel — 1718

Concatenate The Assist Channel With Primary Channel Of Character Data — 1720

Print Concatenated Image — 1722

FIG. 17

FIG. 18

1800

This is the first line of text
followed by many 1802
more lines of text.

This is the first line of text

1910

1902

1912

G15

1904

FIG. 19

This is the first line of text

followed by many

more lines of text.

G1 G2, G3
      1904

1918

1920

1908

1906

1922

Guard digits:

Column Group Size (e.g., 2)

Number Of Guard Digits Per Column Group (e.g., 10)

Number Of Lines Of Text Guarded (e.g., 3)

Vector Of Line Lengths (e.g., {30, 16, 19})

Guard Digits For Each Column Group

(e.g., G1, G2, G3, ... G15)

2002

2004
2006
2008
2010
2012

FIG. 20

POST GRID CODE   2102
Acquire Binary Image

Extract Assist Channel From The Primary Channel — 2104

Decode The Assist Channel — 2106

Decompress The Assist Channel — 2108

Align The Actual Lines Of The Primary Channel Using The Line Lengths In The Vector Of Line Lengths — 2109

For Each Line Of The Primary Channel, Compute A Candidate OCR Interpretation Via A Shortest Path Computation — 2110

Specify A Current Message Vector Using A First Column Group Of The Candidate OCR Interpretation — 2112

Obtain The Guard Digits That Correspond To The Current Message Vector From The Assist Channel — 2114

Decoding Errors? — 2116
No
Yes

Correct The Current Message Vector Using The Guard Digits — 2118

For Each Line Where The Current Message Vector Was Corrected, Reinterpret The Shortest Path Of Computation For The Remainder Of The Line Computed At Step 2110 To Modify The Candidate OCR Interpretation — 2120

Any More Guard Digits? — 2122
No
Yes

Done — 2126

Specify The Current Message Vector To Be The Next Column Group Of The Candidate OCR Interpretation — 2124

FIG. 21

followed by many

followed bj rnany

followed bj rnany

followed bj rnany

2204

2202

2206

2214

2236

2213

2238

2208

1520

2209

2210

2212

2230

1528

1528

1522

2232

2234

1528

FIG. 22

FIG. 23

Receive A Primary Channel Of Character Data — 2302

Select A Current Line Of The Primary Channel — 2304

Let The First Character In The Selected Line Of The Primary Channel Be The Current Character c_i — 2306

Compute A State Change Function $h_i = H(h_{i-1}, c[1,i])$ For The Current Character c_i In The Selected Line — 2308

Last Character In Current Line? — 2310

No

Yes

Compute A Guard Extractor Function $g_i = G(h_i, c[1,i])$ For The Computed State Change Function Of The Current Character c_i — 2312

Let The Current Character Equal The Next Character In The Current Line — 2314

Augment The Assist Channel From The Current Line With The Computed Guard Extractor Functions For Each Character (i.e., $g_1, \ldots g_n$) — 2316

Is The Selected Line The Last Line? — 2318

Yes — 2324

No

Let Current Line Equal The Next Line — 2320

Compute g_n — 2311

Encode Assist Channel — 2324

Concatenate The Assist Channel With Primary Channel Of Character Data And Render On A Hardcopy Document — 2326

FIG. 24

T h i s

$h_0=0$

2406
2408
2410

2404

c_1 8   20
c_2 8 h_1 20 g_1 1   0
c_3 8 h_2 20 g_2 1   1
c_4 8 h_3 20 g_3 1   0
h_4 (i.e., h_n)   20 g_4 h_4

FIG. 25

Acquire Binary Image 2602

Extract Assist Channel From The Primary Channel 2604

Decode The Assist Channel 2606

Systematically Decode Each Line Of The Primary Channel Using The Assist Channel 2610

Done 2612

FIG. 26

`006f50"89d4y560`

2608 (start oval)

2702 Initialize A Priority Queue With An Empty Candidate $C_0=(c'=\emptyset, h'=0, s'=0, k'=0)$ For The Current Line Of The Primary Channel

2704 Remove From The Priority Queue The Candidate With The Best Score And Define It To Be The Current Candidate "C_{best}," Where C_{best}'s Interpretation of the Prefix of the Line Is $c' = c_1'c_2'c_3'...c_{i-1}'$; Let h_{i-1}' Be the Current Hash, Let s_{i-1}' Be the Current Score, And Let k_{i-1}' Be the Location Of The First Unmatched Guard Digit in the Assist Channel

2708 Identify All Extension Characters c_i' Of The Current Candidate C_{best} And For Each Possible Extension Character Score Against It The Remaining Portion Of The Current Line Of The Primary Channel, Obtaining A New Score $s_i' = s_{i-1}' + S(c_i')$.

2710 Compute Values For The State Change Function $h_i'=H(h_{i-1}', c'[1,i])$ And The Guard Extractor Function $g_i'=G(h_i', c'[1,i])$ For Each Of The Identified Extension Characters c_i'

2712 Add To The Priority Queue Candidates $C_{next}=(c' = c_1'c_2'c_3'...c_i'; h_i'; s_i'; k_i')$ For All Identified Extension Characters c_i' That Have A Computed Guard Value g_i' That Equals The Guard Value In The Assist Channel (i.e., g_i) Beginning At Position k_i'

2714 End of Line? (i.e., Does $g_i' = g_n$?) — No (loops back)

2716 All Lines Processed? — Yes → **2612** Done

2718 Advance To Next Line Spacing (No)

2610

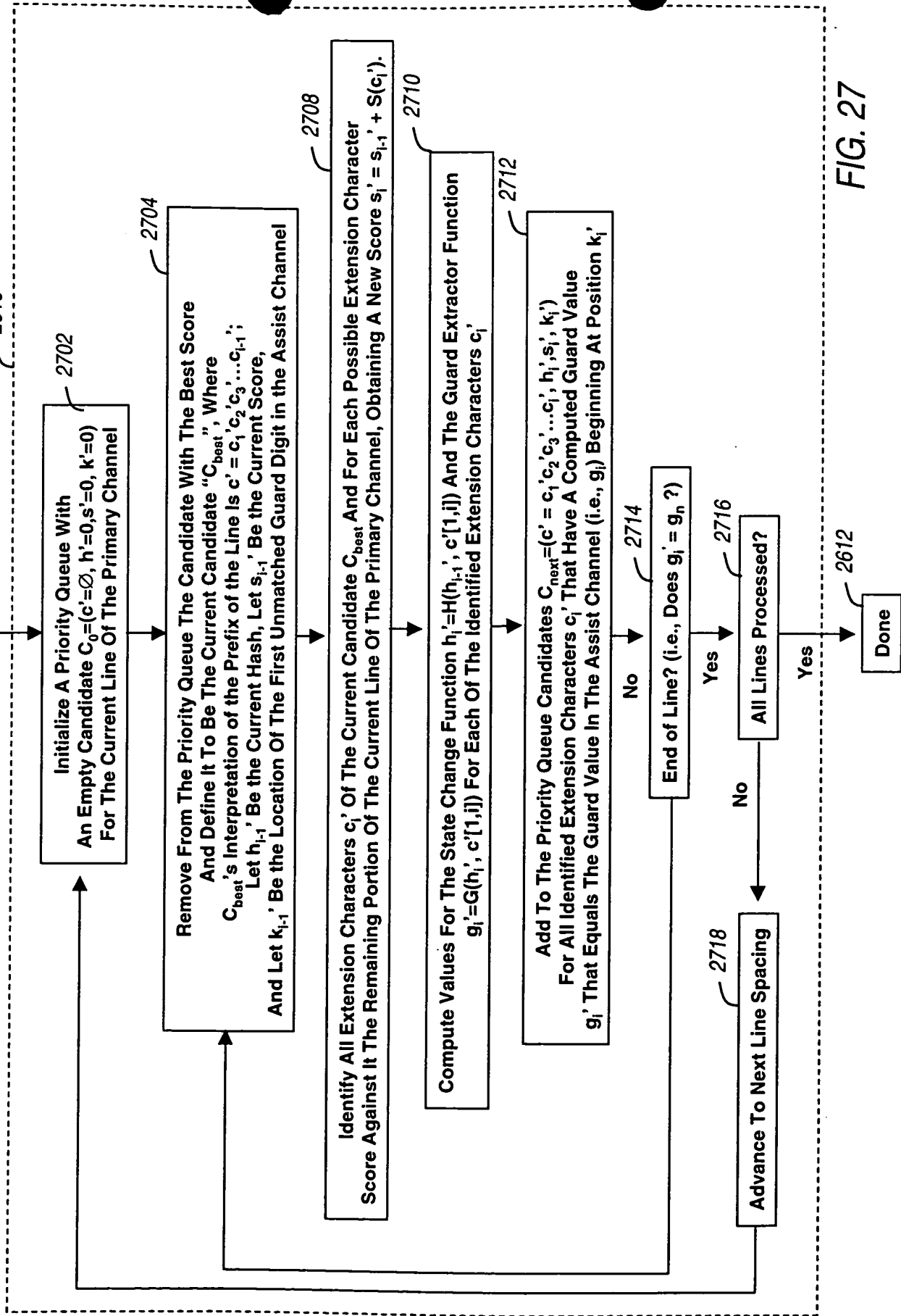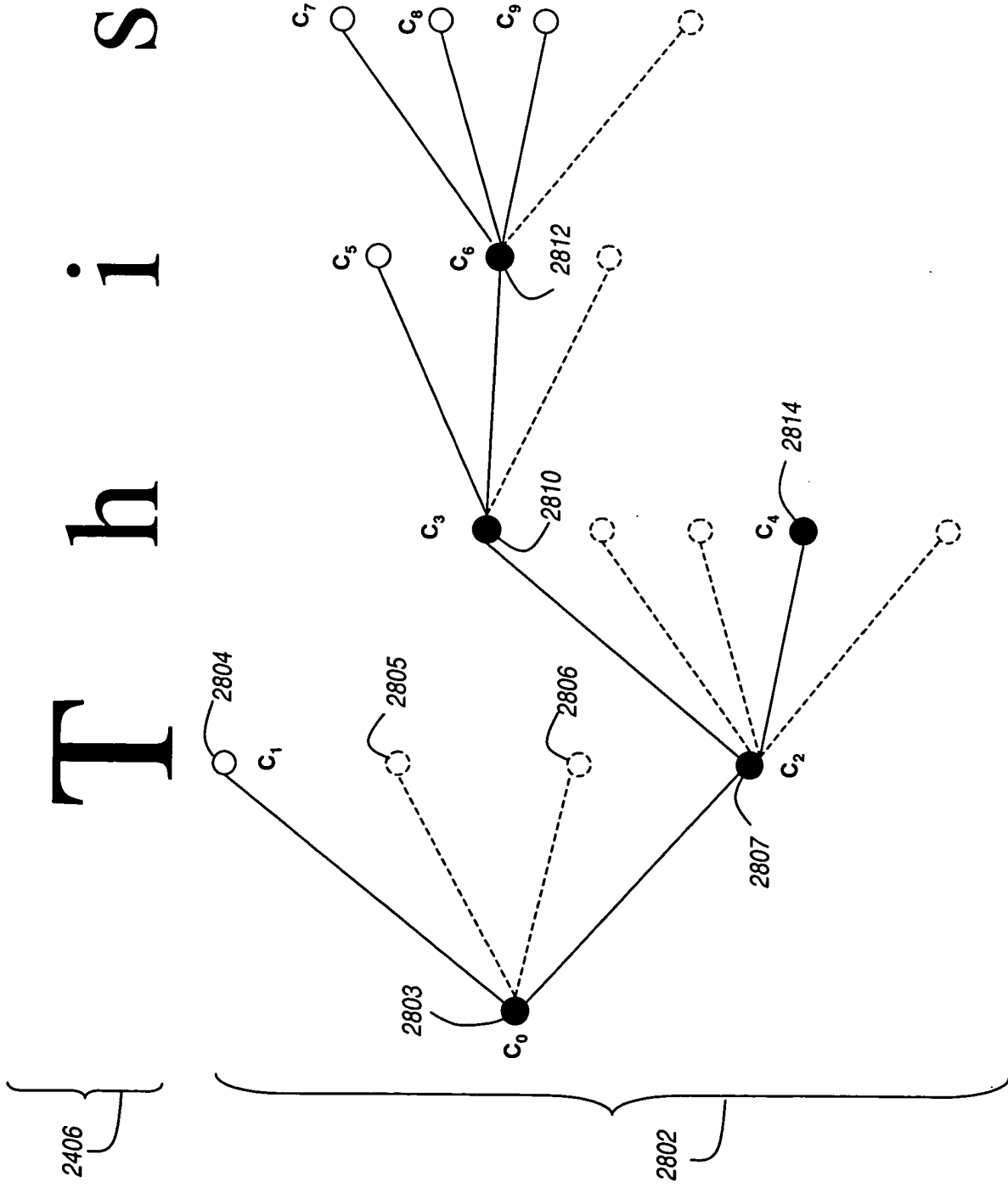*FIG. 27*

FIG. 28

FIG. 29

FIG. 30

FIG. 31

2610

2608

3102 — Initialize A Priority Queue With An Empty Candidate $C_0 = (c' = \varnothing, h' = 0, r' = 0, s' = 0)$ For The Current Line Of The Primary Channel

3104 — Remove From The Priority Queue The Candidate With The Best Score And Define It To Be The Current Candidate "C_{best}", Where $C_{best} = (c' = c_1'c_2'c_3'...c_{i-1}')$; Let h_{i-1}' Be the Current Hash, Let s_{i-1}' Be the Current Score, And Let r_{i-1}' Be the Current State Of The Compressor

3106 — Using r_{i-1}', g_{i-1}', and cd_i' Decompress d_i' From The Assist Channel (d_i' Is The Separation Class For The Next Character), And Compute r_i'

3108 — Identify All Extension Characters c_i' Of C_{best}, Where c_i' Is In The Separation Class d_i', And For Each Possible Extension Character Score It Against The Remaining Portion Of The Current Line Of The Primary Channel, Obtaining A New Score $s_i' = s_{i-1}' + S(c_i')$

3110 — Compute Values For The State Change Function $h_i' = H(h_{i-1}', c'[1,i])$ And The Guard Extractor Function $g_i' = G(h_i', c[1,i])$ For Each Of The Identified Extension Characters c_i'

3112 — Add To The Priority Queue Candidates $C_{nex} = (c' = c_1'c_2'c_3'...c_i', h_i', r_i', s_i')$ For All Identified Extension Characters c_i'

3114 — End of Line? (i.e., Does $g_i' = g_n$?)

No

Yes

3116 — All Lines Processed?

No

3118 — Advance To Next Line Spacing

Yes

2612 — Done

Primary
Channel
112

Rewrite
Module

219

3200

3202

Decoding/
Decompression
Module
212

Decoder
Module

3204

Assist
Channel
118

Bitmap
Primary
Channel
214

Image
Extraction
Module
210

Hardcopy Assist Channel

204

206

FIG. 32

Specific Rules

	Weight
m → rn	medium
d → ol	medium
b → lo	medium
rn → m	medium
j → y	medium
r → v	medium
e → c	medium
c → e	medium

General Purpose Rules

	Weight
Identity Transformation (i.e., Do Nothing) (e.g., a → a)	low
Deletion Transformation (e.g., a → ε)	high
Insertion Transformation (e.g., ε → a)	high
Arbitrary Single Letter Replacement Transformation (e.g., a → ?)	high

3302

3310

3314

3304

3312

3306

3308

FIG. 33

FIG. 34

```
┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐  3400
│                                                       │
│   ┌─────────────────────────────────────────┐  3402  │
│   │ At Every Character Position In The Line   │       │
( begin ) → │ Of Text Define A Node Of A Graph          │
│   └─────────────────────────────────────────┘        │
│                     ↓                                 │
│   ┌─────────────────────────────────────────┐  3404  │
│   │ Add Arcs, And Compute A Score For Each    │       │
│   │ Arc By Matching The String Beginning      │ → ( 2101 )
│   │ At The Node With Each String In The       │       │
│   │ Event Library                             │       │
│   └─────────────────────────────────────────┘        │
│                                                       │
└ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘
```

FIG. 35

followed by many

FIG. 36